ISSN 0956-9979 SEPTEMBER 1999

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **Calling all comparatists:** Our Comparative Review in this issue is for *NT* workstation. Sixteen products took part and this time VB 100% awards were surprisingly few. The review starts on p.15.

• **Independence day:** Kenneth Bechtel sticks his head above the parapet to wave the standard for his fellow independent corporate anti-virus researchers in this month's A Day in the Life on p.11.

• **Everything you ever wanted to know:** about macro viruses but were afraid to ask! Dr Igor Muttik kicks off a new series dedicated to them on p.13.

CONTENTS

COMMENT

Email Viruses: When Threat Becomes Reality

When the macro virus boom shook the electronic security world in 1995, it remained to be seen if they would work hand in hand with email as a means of large scale propagation. During the first half of that year, the challenge became clear. Traditional anti-virus programs were no longer good enough to combat the new menace. Without looking for it, viruses had found the ideal means to spread on a huge scale without being detected, even breaking through the thus far impenetrable firewall barriers. In late 1996, the first large-scale virus infections were perpetrated through email. This soon became the scourge of network administrators – once a virus infected a computer, users unwittingly forwarded infected documents, causing chaos in a matter of minutes.

"... email systems now embody all the dangers of the 'Net."

It was not until the emergence of the Melissa virus that businesses and the media finally came to understand what the most dangerous source of virus infection really is. Up to then, the media limited itself to scaring users by warning them about the dangers of surfing the 'Net or connecting to IRC services. Today, we can prove that most dangers do not lie in the use of the Internet, nor are those that do the most difficult to eradicate. It should be pointed out that everything downloaded by browsers or IRC readers is stored on the hard drive, where any decent anti-virus program can detect and disinfect viruses successfully. Also, some mail readers have incorporated the capacity to send and view messages in HTML, so that email systems now embody all the dangers of the 'Net.

Very much to the contrary of what was believed up to now, files received through messages, mailing lists or newsgroups are more difficult to control. We are passive subjects of attack, since infections are not produced by browsing through dangerous sites or by chatting over insecure channels, but by receiving unsolicited messages from unknown users. To complicate things further, it is easier to identify the owner of a potentially dangerous Web site than it is to track down a user who sends infected documents or files. Happy99, Melissa, ExploreZip and the like were designed to take full advantage of this situation by getting friends and colleagues to send us infected messages. Thus, trust comes into play as another major factor in the infection process. In short, the problem of viruses in email messages was not given the attention it deserved until it was too late. These viruses are invisible to traditional anti-virus programs, and can get through security systems such as firewalls. They have tremendous replication potential, can be unknowingly converted into passive accomplices of attack and incorporate all the risks of other systems. Furthermore, they can include executables, files containing macros, HTML files, OLE objects, etc.

In recent months, and especially since the release of *Windows 98*, a new means of virus proliferation has sprung up through the use of the *Windows Scripting Host*. This has been created to replace traditional BAT files, but the new files are programmed mainly using JavaScript or Visual Basic Script. Their most important characteristics are ease of programming, total access to the system and total access to OLE objects through automation. To date there have been no reports of major infections taking place using this system, but it will become an important source of infection in the near future. Melissa already uses *Outlook* and *Word* OLE automation to infect and spread.

An efficient anti-virus program must be able to scan and disinfect viruses in all major messaging systems that attach files to messages. For corporate GroupWare solutions, such as *Lotus Notes* or *MS Exchange Server*, which can be used as Internet mail servers, it is vital that the product include a permanent protection feature for Public and Private Information Stores and also for the connectors that serve as a means of transport for messages *to* and *from* the Internet, not to mention the Personal Folders stored on workstations. For *Exchange Server*, we would be talking about the Message Transfer Agent (MTA), which includes connectors such as Internet Mail Connector or the X400 Connector. AV software should come with permanent protection systems that prevent viruses from ever reaching hard drives. Of key importance are those products which act as the computer's first line of defence filtering email and news packages sent through TCP/IP as well as the specific permanent protection features designed for each type of mail system.

Jesus Valbuena, Panda Software, Spain

NEWS

Symantec BlOws It Off

On 2 August Web surfers were encouraged to visit the official *Symantec* Web site as the company celebrated a fourth consecutive month as top software retailer in the UK, outselling *Microsoft* in the month that *Office 2000* shipped. That very day a group of five hackers calling themselves BlOw claimed that their Worm, h3r3, had been infecting the same site for the past two months.

While *VB* is the first to agree that there is no such thing as bad publicity, this hack is timely if nothing else. In a rare show of solidarity other relieved anti-virus vendors have been quick to sympathise with *Symantec*, while taking the opportunity to point out extra features and additional strengths to their own particular product lines.

While the *FBI* investigates, *Symantec* denies that any infection ever took place and can only praise its staff for a 'speedy' reaction to the incident and a hasty recovery of business as usual as befits Britain's number one software retailer. Talk about containment. *VB* suggests that this incident is gone but not forgotten. In the AV marketing world, a little complacency goes a long way ▮

Anyone for Monopoly?

The recently discovered VBS/Monopoly Worm requires the use of the *Windows Scripting Host* to ensure propagation. It also emails information, which could later be used for spamming, from your computer to several addresses. Like Melissa, the Worm appears in your mailbox from someone you know with the subject line 'Bill Gates joke' and the body of the text message saying 'Bill Gates is guilty of monopoly. Here is the proof :-)'. Attached is the file MONOPOLY.VBS.

When MONOPOLY.VBS is executed it performs several tasks in order. MONOPOLY.VBS is interesting in that all the filenames and the data for the created files are encoded and there is a small function which decodes these strings at runtime. When the MONOPOLY.JPG file is opened via a Jscript, a joke Monopoly board picture is displayed.

When the MONOPOLY.WSH file is executed, so is the file MONOPOLY.VBE. Running this script causes an an email to be sent, using *Outlook*, to all entries in all address lists. This has the same subject line mentioned above and the same 'smiley face' message. The MONOPOLY.VBS file is attached. We are now back to where we started. Messages are automatically deleted from the sender's outbox.

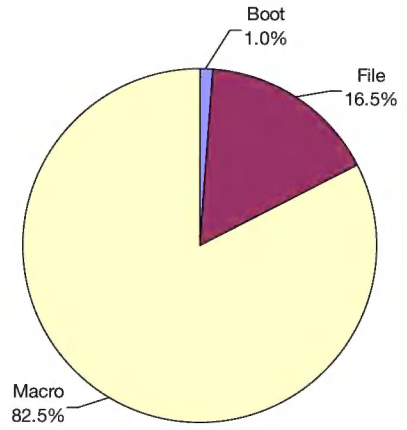
Once the messages have been sent, Monopoly sends an email which is copied to several addresses. Finally, the script sets a registry entry which ensures emails are only sent once. For full details see http://www.virusbtn.com/ ▮

Prevalence Table – July 1999

Virus	Type	Incidents	Reports
ColdApe	Macro	299	26.82%
Win32/Ska	File	160	14.35%
Ethan	Macro	129	11.57%
Laroux	Macro	106	9.51%
Marker	Macro	97	8.70%
Class	Macro	94	8.43%
Melissa	Macro	53	4.75%
Tristate	Macro	33	2.96%
Footer	Macro	30	2.69%
CAP	Macro	29	2.60%
Win32/ExploreZip	File	11	0.99%
Win95/CIH	File	8	0.72%
Story	Macro	6	0.54%
Chack	Macro	4	0.36%
Compat	Macro	4	0.36%
Concept	Macro	4	0.36%
Extras	Macro	4	0.36%
Groovie	Macro	4	0.36%
Form	Boot	3	0.27%
Nono	Macro	3	0.27%
PSD	Macro	3	0.27%
Others [1]		31	2.78%
Total		**1115**	**100%**

[1] The Prevalence Table includes a total of 31 reports across 22 other viruses. A complete summary can be found at http://www.virusbtn.com/Prevalence/.

Distribution of virus types in reports



LETTERS

Dear Virus Bulletin

Bad Excuse for a 'Good' Virus

The idea of the 'good' or 'beneficial' virus is as old as the idea of a virus itself. Derived from theoretical works, defined and described by the mathematical models of computer viruses, the possibility of the existence of a 'good' virus has been proclaimed and advocated by scientists like Dr Fred Cohen for the last ten years.

In real life the same idea usually serves as the lame excuse for creating yet another useless virus. As someone who has to deal with real viruses on a daily basis, I've seen many ideas and efforts to write a 'good' virus (a virus battling another 'bad' virus or a virus exposing security weaknesses, or a virus 'teaching users a lesson', or a military virus wreaking havoc in enemies' systems, etc) that have ended up in the creation of programs no different from thousands of other viruses. It is also important to understand that these viruses are perceived by users as just the same as all the others – unwanted.

The discussions about 'is the idea of a good virus viable and worth pursuing in the real world?' have burst onto Internet forums so often in the past decade that users are advised to get acquainted with the arguments so far before triggering another flame war.

After reading the letter from Jeremy (see *Virus Bulletin*, August 1999, p.5) I had the impression that its author either was not aware of earlier discussions or had purposely decided to ignore some important arguments against so called 'useful' viruses.

Below I list a number of issues that ought to be considered in this scenario:

- Programs that run a self-checking program before executing might not run after being infected with a 'good', or any other type, of virus.
- The history of the anti-virus industry is illustrated with many disastrous cases when this type of active protection was applied to programs that had already been infected.
- Modifications to some programs will nullify any warranty or technical support a user was entitled to before the 'protection' was introduced.
- Introducing an active 'good' virus takes some extra system resources.
- A purposely infected machine forces a user to rely solely on this one protection since most other anti-virus schemes will detect and try to eradicate a 'good' virus.
- Unless an infected/protected computer is isolated, working on such a system creates a risk where a 'beneficial' virus may escape into the wild.
- Jeremy's proposal for users to buy a 'good' virus in a box (I guess, through legitimate distribution channels) will break many existing anti-virus laws.
- Jeremy's protection will also be ineffective in cases when a 'bad' virus infects programs without modifications to their code. This is also true if, when infecting a program, a virus corrupts or disables the already installed 'good' virus.
- The proposal does not even mention macro viruses which are the single most important part of the current virus problem.

While I'm clearly against the idea of writing any kind of virus, the final word, however, belongs to the users. Is the PC community willing to embrace the idea of 'beneficial' viruses and accept 'good' viruses infecting their systems in order to stop 'bad' viruses? I think not.

Jakub Kaminski
Virus Research Manager, Computer Associates
Australia

To Disclose or Not to Disclose?

Today, one finds the anti-virus industry pondering this question more than it ever did in the past. Previously, the question of full disclosure only related to viruses and the answer was clear. The industry is in full agreement – anti-virus companies, researchers, and testers do not give out samples of viruses to the general public. Even stricter, generally samples are not even passed to other researchers until a sufficient trusted relationship exists. To me, this is reasonable. These are the creations we are trying to prevent from spreading. Providing full disclosure of viruses or virus code to the general public does not help prevent the spread of viruses in any way.

However, what happens when Russ Cooper of *NTBugtraq* wishes to release a demonstration exploit of a bug in *Microsoft Office 97* which can potentially cause malicious damage? Well, for anti-viruses researchers, the answer becomes less clear. The official standpoint of the *Symantec AntiVirus Research Center* (*SARC*), and my personal belief, is not to release such a demonstration exploit. However, other researchers disagree. In fact, in the past, security vendors have released demonstrations of similar exploits (*Finjan* and the Russian New Year exploit). So why is the stance of anti-virus researchers, regarding the distribution of a demonstration of the recent *Office 97* exploit, any different from their current beliefs regarding the disclosure of viruses and Trojans? Intent? If the idea is that distribut-

ing such code forces people to patch their products and brings attention to the security issue, I ask what is different about viruses and Trojans?

One could claim that we should release non SR-1-capable *Word 97* viruses to the general public to convince them to apply the SR-1 patch to *Word 97*. Better still, perhaps we should cause corporations to put pressure on *Microsoft* to create non-macro capable versions of the product. While some might exclaim 'Hear, hear!', I guarantee you AV researchers would balk at the claim and for good reason.

Then, tell me, why is releasing the *Office 97* exploit code justified, even if it is just a demonstration? If a virus was created as a demonstration (WM/DMV), should that code be released? If you claim the demonstration tool is not actually malicious, then I counter and ask, 'Do you think we should release the source code of an intended virus or Trojan, then?' After all, intended viruses and Trojans do not work and thus, are not malicious.

Clearly, the release of intended virus code is frowned upon by the anti-virus industry because with a few tweaks the intended virus code could easily become live virus code – as can a demonstration tool for the *Office 97* exploit. Let us not enable those from whom we are trying to protect the general public. I personally believe many items of full disclosure are warranted. In fact, I believe Russ Cooper provides an invaluable service with *NTBugtraq*.

However, is it necessary to release demonstration code? Is it not enough just to release details of how the exploit works? This prevents the 'wannabe' hacker from simply modifying code to change a seemingly harmless exploit into a Trojan in minutes, while still providing critical details allowing IT administrators, other product vendors, and security folks to verify the claims and allow them to determine if and how they will be affected.

The AV industry is here to help stop the spread of malicious code. Let us continue that with the disclosure of potential exploits to warn computer users. Let us not approve the actual creation of malicious code that utilizes such exploits.

Eric Chien
Senior Researcher, SARC EMEA
Leiden, The Netherlands

Throwing Down the Gauntlet

Over the past several years I've noticed that your anti-virus software comparative findings sometimes contradict those of certification organizations. Products which are certified to detect 100% of viruses found to be in the wild are shown in your tests to be not quite as effective as the certification seems to indicate.

While I don't have time to go back and research all of the discrepancies, I think it would be interesting for your readers to see a comparative of the results of your tests against certified products. I would also be interested in seeing if there are times when your VB 100% awarded products have been shown by any other certifications to be not quite as effective as your tests might indicate.

This is of particular interest to readers as you undertake on-access testing as part of your VB 100% Certification. Are you up to the challenge? Are they?

Dr Richard Ford
Consulting Editor, Virus Bulletin
USA

Taking Issue

Suggestions in Mr Nachenberg's letter from the August issue are too outrageous to be left uncorrected. Further, I suspect that most of those qualified to comment may avoid doing so because of concern their comments may be seen as representing their (employer's) vested interests.

Nachenberg's letter was written, if not published, at the height of a *Symantec* publicity campaign to warn its users that its product could not protect against the latest, freely-spreading Melissa variant. The official line was not that blunt, but that was the essential message. As Nachenberg explained, some file types under some OSes will be opened based on their content, if their file extension is not associated with an application. In fact, *Microsoft* is moving more and more to this model – PCs running *Windows 98* have more 'holes' of this nature than those running *Windows 95*, and *Internet Explorer 5* introduces even more again.

From the user's perspective, this is probably desirable, but it is a nightmare for extension-based scanners. However, the blame does not lie solely beyond anti-virus vendors. Shortly after I took the helm at *VB*, I informally warned many anti-virus developers of this 'problem' with renamed *Word* and *Excel* files. No viruses took advantage of it, but it struck me as important, something developers should look into, and not something to publicize. Some responded that their on-access scanners had 'intelligent typing', looking briefly at all files to decide whether they were of types that needed 'proper' scanning. Most on-demand scanners still used an extension list to decide the files to scan by default, although some were also moving to intelligent typing.

Other solutions are also available. For example, developers could build mechanisms into their update procedures to allow virus definition files to carry configuration changes (such as desirable alterations to the extension list). Again, I have offered several such ideas to many developers, but few show any sign of adopting them. Regardless of the *Symantec* position on these minutiae of product design, Nachenberg's claim that 'anti-virus products neglected to scan these files' is puffery at best. Its strong implication that *all* competing products suffered the same weakness as his own should not be left unchallenged.

Nick FitzGerald
Computer Virus Consulting Ltd
New Zealand

VIRUS ANALYSIS 1

If the CAP Fits

Nick FitzGerald

It is almost three years since the release of WM/CAP.A. In that time it toppled WM/Concept from its place at the top of many virus prevalence lists. Despite 'competition' from several other macro viruses, and brief surges from the likes of Win95/CIH and Win32/Ska, it held the top spot on the *Virus Bulletin* Prevalence Table for 18 of the 29 months since first listing in February 1997, and has maintained a top-ten position since. So, what makes CAP tick? Why has it been so 'successful'?

CAP's Structure

In its simplest form, WM/CAP.A consists of ten WordBasic macros. There are three auto-macros (AutoClose, AutoExec and AutoOpen), six system macros and the one that gave it a name, CAP.

Two of the system macros hook *Word's* FileTemplates and ToolsMacro functions and are detailed later, in the Stealth section. The other system macros hook perhaps the most important internal *Word* functions. Generically, these are FileClose, FileOpen, FileSave and FileSaveAs, and are significant, because it is impossible to do anything useful in *Word* without using them.

Macros with names matching internal *Word* functions run in preference to those functions. Such macros can still call the internal function, should they need to use its functionality. Thus, hooking *Word's* critical file functions allows CAP to infect even the most cautious user. When CAP was a 'new' virus, someone who had disabled auto-macros *and* habitually held down one of the Shift keys while opening files (to disable any AutoExec macros), would become infected, because once an infected document was open, use of *Word's* basic file handling functions would run the virus' code.

Language Independence

CAP was certainly not the first macro virus to take advantage of *Word's* precedence rules. However, what made CAP unique was the mechanism its writer devised to overcome language sensitivity problems inherent in that approach.

Localization of the *Word 6/95* program, affected the names of its system functions. A function can have different names in different language versions of *Word*, and many WordBasic viruses have language version limitations. An English macro that *calls* the FileOpen function will work fine in the German version of *Word*, as the p-code for FileSave is the same as for DateiÖffnen. The problem arises when a macro is *named* after an internal function and the virus depends on usurping it. As macro names are literals, a macro called FileSave is just that under any language version – it does not become the macro DateiÖffnen under German *Word*.

The *CARO* naming standard allows for a language modifier after the variant indicator. For example, WM/Boom.A:De was in the wild for some time, but only replicates under the German version of *Word* and, similarly, WM/TWNO.A:Tw was in the wild but only replicates under Taiwanese *Word*.

The CAP Macro

WM/CAP saves infected documents as templates, as its host platforms only support macros in template files – a restriction that *Microsoft* saw fit to remove from versions of the product subsequent to *Word 95*. As *Word* does not warn users that a file's internal format does not match what may be expected from its name, it is likely most users would be unaware they were opening files that could carry macros. Users of *Word 95a* should receive the standard 'macros and customizations' warning that was introduced in that release.

The CAP macro's main sub-routine is an otherwise empty stub, containing a four line comment:

```
'C.A.P: Un virus social.. y ahora digital..
'"j4cKy Qw3rTy" (jqw3rty@hotmail.com).
'Venezuela, Maracay, Dic 1996.
'P.D. Que haces gochito ? Nunca seras Simon
    Bolivar.. Bolsa !
```

This suggests the virus' writer is the teenager who later wrote Win32/Cabanas (see *VB* November 1997, p.10).

Apart from its null main routine, the CAP macro is the engine room of the virus. It consists of five sub-routines, with the bulk of the code in S, the main infection routine. An understanding of the CAP.S routine is crucial as it is called, directly or indirectly, by all the other active macros and sub-routines. S first sets a 'resume next' error handler so all WordBasic errors are quietly ignored. An array of strings is then built,establishing a canonical list of 'core macros' for the virus. It includes only the English versions of the language dependent functions, even when run under different localized *Word* versions.

Macros in the active template are then checked for descriptions beginning 'F%'. CAP's macros meet this criterion; any that do not are deleted. The names of macros passing the descriptor test are compared with those in the canonical list of names, and matches increment a counter. This is all repeated with the normal template and a second counter.

CAP.S expects a single parameter which is checked next. If it is a null string the infection process is skipped, otherwise the parameter is the target's filename and the two counters described above determine if the normal template has a full complement of core CAP macros. If not, it is infected.

Infecting the normal template is complex, as this is where CAP 'localizes' its language-sensitive macros. First, the S sub-routine disables 'prompt to save normal template' and enables fast saves and auto-save. Each macro in the source file is copied to the normal template and, except for ToolsMacro, is made execute-only. ToolsMacro is left editable so its description can be changed. After its 'F%' marker is a generation count, which is incremented next.

Then the 'local' names for FileClose, FileOpen, FileSave and FileSaveAs are obtained. This is achieved by extracting the function names associated with specific menu entries with the MenuText and MenuItemMacro functions and by assuming the menu and item positions for these functions are the same across versions. If the local names and macro names do not match, the macros are copied to the normal template with the localized names. Thus, under non-English *Word* versions, CAP consists of the ten core macros, plus copies of FileClose, FileOpen, FileSave and FileSaveAs. In fact, CAP-infected documents that have visited several different localized *Words* will contain the additional localized macros from each *Word* version.

Infection of a document is much simpler. Following the purge of any non-CAP macros from the host and the normal template, the host's format is checked. If it is *Word* template or document, or RTF, all macros in the normal template are copied to the host, its format changed to template and the file saved. Thus files with RTF extensions can be CAP-infected *Word* document format files.

Other sub-routines in the CAP macro are FC, FO, FS and FSA. The four file function macros already discussed are simple routines that set 'resume next' error handlers then call their obvious acronymic partner from this list. Apart from FSA, these routines simply perform a call to the *Word* function matching their expanded acronym and the CAP.S routine. CAP.FSA is more complex.

If the file being saved is not a template, CAP.FSA calls the internal FileSaveAs function then passes the file to CAP.S for infection. Templates cause serious problems for most WordBasic viruses. The FileSaveAs function insists on saving templates to the user template folder listed in *Word's* File Locations option, and as *Word 6/95* only supports macros in templates, infected files have to be saved as templates. With subsequent use of the SaveAs command on an infected file, all but the least observant user is left wondering why they can not save their documents anywhere other than the templates folder.

CAP.FSA resolves this in a novel way. Should it intercept a FileSaveAs on a template, it creates a new document based on the template, sets the file name to the template's then calls the internal FileSaveAs function – the user may save the file wherever they like and as any format. Observant users will notice 'strange' changes to the document name in *Word's* title bar while this happens, and that saving seems slower than usual. That said, the approach is quite successful in overcoming the problem described above.

Auto Macros

Normally, CAP would gain control through its AutoExec macro. Template files containing AutoExec macros have that macro executed when opened in *Word 6/95*, even if auto-macros have been disabled. AutoExec enables auto-macros and calls CAP.S with a null string argument. AutoOpen and AutoClose are similar, but do not enable auto-macros and pass the current filename to CAP.S.

Stealth

None of the attempts to hide CAP's presence are wholly successful, and some are downright flawed. Under English *Word*, the FileTemplates and ToolsMacro commands are disabled by the 'do nothing' macros with matching names. Strangely, CAP.S does not attempt to localize these macro names in a similar manner to the main File macros.

When infecting the normal template, CAP.S also searches the File and Tools menus for items whose function names contain 'Macro'. Although the menus to search are located in a language independent way, the search string is not. If located, the menu item and the one below it are deleted. In English *Word*, the Macro and Customize items are deleted from the Tools menu and the Macros item is deleted from the File menu displayed if no files are open. Localized *Words* where 'Macro' is spelt differently (e.g. 'Makro' in German), will not be affected.

The macro organizer can still be reached through Format, Styles, even in English *Word*. The menu deletions and 'do nothing' macros have little or no effect in other versions.

Summary

CAP may be the most widespread virus ever. Its enduring 'success' is partly due to the success of *Word* (itself due to *Microsoft's* localization efforts) and the extensive use of 'old' versions of the program in regions that cannot afford hardware upgrades as regularly as the West's 'upgrade or perish' mentality recommends. It has also been successful because it overcame the FileSaveAs problem and does not draw attention to itself with destructive payloads.

WM/CAP.A	
Alias:	None known.
Type:	Word Basic virus infecting *Word 6* and *Word 95* files. RTF files are saved as infected *Word* files with RTF extensions.
Self-recognition in Files:	Count and name check of macros whose description begins 'F%'.
Payload:	User macros in host files are deleted.
Removal:	Delete the virus' macros from affected files. Be sure to check RTF files.

VIRUS ANALYSIS 2

Picturing Harrier

Eugene Kaspersky
Kaspersky Lab

Win32/Harrier is a polymorphic, parasitic Win32 virus. It stays resident in the *Windows* memory and infects accessed PE EXE files. The virus manifests itself with video effects: displaying message windows, changing text strings in program messages and creating a BMP file. It has bugs and in some cases infected programs are terminated by *Windows* with a standard error message.

Harrier is large: the length of affected files grows by about 100 KB. About 10 KB of virus code in infected files is occupied by polymorphic decryption loops, and about 90 KB by real virus code. Fortunately, about 60 KB of virus code is occupied by data – text strings (virus messages etc), a BMP file image, etc. However, the rest is about 30 KB of assembler code – about 8,000 assembler instructions.

The virus' polymorphic engine may be classified as an average one, but it generates between nine to seventeen decryption loops. These decrypt virus code, mixed with many junk instructions, layer by layer, and as is mentioned above, generate about 10 KB of polymorphic code.

Having been decrypted, the virus code appears in very strange form. Each virus instruction is followed by a JMP opcode that passes control to another place in the code. The virus assembler instructions seem to be randomly mixed in the virus code and linked by JMP commands. Instructions of any virus routine may be found at any position in the virus code. They may be as far away from each other as several kilobytes. Fortunately, the virus has no mutating engine in its code – the sequence of virus instructions stays the same in each infected file. It seems that the virus author used some special tool to mix source assembler code before compiling it to the first-generation EXE file.

Installation

When the infected file takes control and the decryption loops restore the code to its original form (not encrypted, but JMP-linked), the installation routine takes control. This routine uses calls to several *Windows* functions. Unlike the majority of other currently known, parasitic *Windows* viruses, this one does not scan the host file, Import table or Kernel32 Exports for their addresses.

Harrier uses a new way to access them. While infecting a file, the virus patches the program Import table and adds its own table. This table is built so that *Windows*, while loading infected files, links the virus code with the addresses of necessary *Windows* functions.

The virus then relocates its own data. There are 288 instructions in the virus code that use direct addresses to access virus data or call virus subroutines. These addresses have to be fixed to point to authentic virtual addresses where the virus code is placed by *Windows*. The virus uses a silly loop to do that, and then installs *Windows* API hooks and returns control to the host program.

To install their hooks the virus scans the host file Export table and patches necessary fields with virus hookers' addresses. So it is able to hook only those functions that are used by the host program, and the virus hookers get control only if the host program calls corresponding functions. The second feature of Harrier's 'memory residence' is that it is 'per-process memory resident'. The virus copy is active only during the period the host file is run, and is moved out of *Windows* memory when the host file exits.

The virus hooks 31 *Windows* API functions: 12 KERNEL32 functions, four SHELL32, two COMDLG32, eight USER32, and five from the GDI32 library. All KERNEL23, SHELL32 and COMDLG32 hooks are used to infect executable files accessed by these functions: open/create file, move/copy, execute/load, find file, etc. All other hooks (USER32 and GDI32) are used in Harrier's trigger routines.

Infection

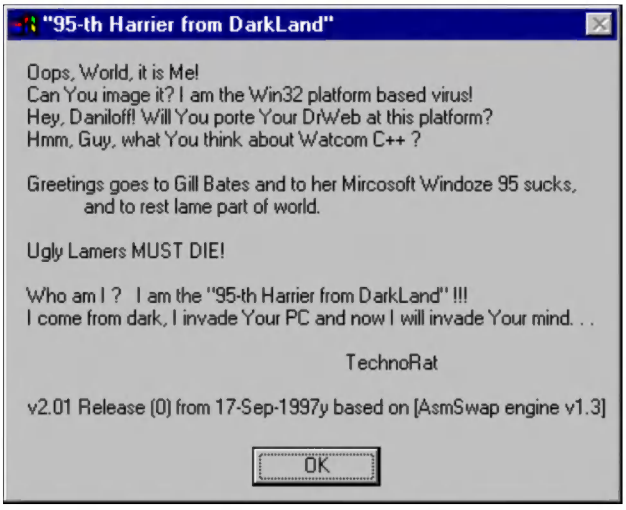
While infecting a PE EXE file, the virus parses its internal file format, creates one more section at the end of the file to which it writes its encrypted text. The virus section is continued by the virus' Export table that is used to link its code with necessary *Windows* API functions when an infected file is executed. Since the virus has its own Export table, it modifies the pointer to it in the PE header. Harrier also pays special attention to the original host file's Export table. To save it, the virus moves necessary data from there to the file end and appends it to its own Export table. As a result, when *Windows* loads infected files, it processes both the virus' and host's Export tables.

To link its section with victim file body, the virus modifies necessary fields in the PE header. It does that very accurately, and as a result, in most cases it does not cause errors when infected files are loaded, even under *NT*.

The virus detects already infected files by a stamp that is saved in the LastWrite date and time stamp file. This ID value is not constant and depends on other fields (the virus Rol/Ror/Xor-es five of them to caclulate the ID).

Trigger routines

While installing as memory-resident, the virus calls three of its trigger routines. The first of them checks the system environment and, dependent on that, turns the virus to the 'debug mode'. The second, depending on the system time's seconds value, displays this dialog box.



The last line depends on the virus random counter (that depends on the system date and time). In one case out of sixteen, the OEMINFO.INI and OEMLOGO.BMP files are dropped to the *Windows* system directory.

The OEMLOGO.INI file contains the 'HARRIER from DarkLand' logo and a text string in two sections, including attributing credits by the author, TechnoRat, and the text:

```
'Today the virus is not the virus,
but the part of operating system …'
```

This BMP file and 'General' sections are shown in the 'System Property' window when MyComputer/Properties is selected. The virus 'Support Information' is displayed when the corresponding button in the same window is pressed.

The virus 'debug mode' is activated when the system environment contains a specific string (for instance, 'Variable=Value' is set by a 'SET=' DOS instruction). This string has 19 symbols and is detected by the virus by using a silly CRC loop which 'compresses' the string to four bytes, so there are several million 'readable' variants of it.

When the Harrier virus' debug mode is on, it displays a message box which says 'entering to DEBUG mode'. Subsequently, the virus displays this dialog box on each infection, requesting permission to infect a file. If one presses 'OK' the virus runs its infection routine. However, if one presses 'Cancel', it displays yet another message box – 'Infecting aborted by Creator!' – and exits.

As mentioned above, the USER32 and GDI32 hooks are used by this virus in its trigger routine – the virus changes the texts that are displayed, or outputs its own messages. When an infected application calls to WinHelpA function for the sixteenth time, Harrier displays its own dialog box instead of calling a *Windows* function. It says:

```
"95-th Harrier from DarkLand"
God will help! ;-)
```

On any MessageBoxA call the virus checks the system time and depending on it replaces the original text in the dialog box with one of six variants:

```
System malfunction!
VXDs rings overcrossed!
CPU mode thunking error!
CPU overclocked, cooler device emergency!
Help subsystem is damaged!
Attention! Bugs inside computer, use
SoftIce.
```

On other hooked calls the virus scans the text for four variants of substrings and replaces them with its own versions:

```
MICROSOFT  -> MIcrOSOFT
WINDOWS    -> WINDOwS
BILL GATES -> Gill Bates
HARRIER    -> Oh! Guys! Is it about me?
```

Conclusion

The Win32/Harrier virus was first discovered at the beginning of 1999. Even now, it is not known to be in the wild and thus many users may never come across it.

However, it is representative of the developments that are being made in the creation of this kind of virus. Taken with the analysis overleaf of Win32/Bolzano, we can soon expect more and more viruses for this platform.

Win32/Harrier

Aliases:	Win95/Harrier.
Type:	Resident Win32 PE infector.
Self-recognition in Files:	By a date and time stamp that is saved in a file.
Payload:	Displays one of several dialog boxes depending on the system time's seconds value. Depending on the virus' random counter drops the files OEMINFO.INI and OEMLOGO.BMP shown in the 'System Property' window when MyComputer/Properties is selected.
Removal:	Replace infected files from backups or originals.

VIRUS ANALYSIS 3

Bolzano Bugs NT

Péter Ször
Symantec

The first genuine Win32 virus – Win32/Cabanas – appeared at the end of 1997 (see *VB*, November 1997, p.10). From early July 1999 onwards we have been analysing six or seven of these viruses a week – a new 32-bit *Windows* virus almost every day! There are more than 200 variants now and shipment day for *Windows 2000* is getting very close.

Win32/Bolzano is a new virus that replicates under *Windows 95* and *NT*, infecting Portable Executable applications with EXE or SCR extensions. The virus does not infect if the size of the host program is less than 16 KB. We have received four different variants of Bolzano so far. A, B and D are very buggy, but the C variant is more stable. The size of D variant is the longest at 2716 bytes, but infected files will grow by at least 4 KB.

From the virus replication point of view there is nothing too remarkable about Bolzano. It is a simple, direct action appender. It adds its code to the end of the last file section and modifies the entry point of the program to point to the virus body (A, B and C variants). The D variant does not modify the entry point of PE files; instead, it searches for 12 possible CALL instructions inside the code section of the host and hooks the randomly selected CALLs to the entry point of the virus.

Fortunately, the D variant is not polymorphic; if it were, detection would be very difficult. The virus creates a thread in the infected process for itself and replicates in the background while it executes the host program (main thread). Therefore the user will not easily notice any delays. The B, C and D variants not only replicate but attack the *NT* file security system by using a new strategy which is likely to be used by other *NT* viruses in the future. This attack works on any version of *Windows NT* from 3.50 up to 4.0 with each service pack. It does not work on any betas of *Windows 2000*, but it remains feasible.

In order for the virus to attempt the attack, it needs administrative rights on an *NT* server or workstation during the initial infiltration. Therefore, it is not a major security risk, but still a potential threat. Viruses can always wait until the Administrator or someone with equivalent rights logs on. Then Bolzano has the chance to patch NTOSKRNL.EXE, the *NT* kernel, located in the WINNT\SYSTEM32 directory. The virus modifies just two bytes in an undocumented security API called SeAccessCheck that is a part of NTOSKRNL.EXE. In this way, the Bolzano virus is able to give all users full access to each file, regardless of its protection, whenever the machine is booted with the modified kernel.

This means that a Guest with the lowest possible rights on the system will be able to read and modify all files including those which are normally accessible only by the Administrator. This is a potential problem since the virus can spread everywhere it wants to regardless of the access restrictions on the particular machine. Furthermore, after the attack no data can be considered protected from any user. This is because the modified SeAccessCheck API is always forced to return 1, instead of 0 or 1. 1 means that the particular user has the necessary rights to access a particular file or directory placed on an NTFS partition while 0 means the user has no access. SeAccessCheck is called each time the file access rights should be checked.

Unfortunately, the consistency of NTOSKRNL.EXE is checked in only one place. The loader, NTLDR, is supposed to check it when it loads NTOSKRNL.EXE into physical memory during machine boot-up. If the kernel gets corrupted, NTLDR should stop loading NTOSKRNL.EXE and display an error message even before a 'blue screen' appears. In order to avoid this particular problem, Bolzano also patches the NTLDR so that no error message will be displayed and *Windows NT* will boot just fine even if its checksum does not match with the original.

Since no code checks the consistency of NTLDR itself, the patched kernel will be loaded without notification to the user. Since NTLDR is a hidden, system, read-only file Bolzano changes its attributes to 'archive' before trying to patch it. The virus does not change NTLDR's attribute back to its original value after the patch. Bolzano's B, C and D variants delete the contents of the \WINDOWS\Cookies and \WINNT\Cookies directories. Probably the virus writer wants to introduce the virus onto a machine he was using to cover where he was Web-surfing.

It is very likely that we are going to face other viruses that will be able to infect the *Windows NT* kernel and load themselves into the kernel memory area by using a similar attack. This would leave very little business for anti-virus companies that do not have an on-access, *Windows NT* driver-based scanner.

Win32/Bolzano

Infects:	Portable Executable files.
Self-check:	Time/Date stamp – not reliable, causes double infections.
Trigger:	Deletes \WINDOWS\COOKIES and \WINNT\COOKIES directory and patches NTLDR and NTOSKRNL.EXE.
Removal:	Delete infected files and use backups.

A DAY IN THE LIFE

The Vendor's Friend

Kenneth Bechtel
Independent corporate anti-virus researcher

Independent corporate researchers are often the forgotten link in anti-virus research, classified by some peers as neither engineers nor researchers and by the LAN staff they support as something akin to 'them who dabble in the black arts'. More and more we are becoming the link between the first discovery of a virus and the anti-virus product developers. Often, good researchers are the difference between a low impact occurrence and a critical incident. However, if they are corporate employees, it is usually frowned upon for them to publish, or 'go public' in any way.

While I cannot speak for everyone, I certainly can relay my experiences supporting corporate America. My anti-virus experience dates back to 1988 and several companies, organizations and military service but I shall concentrate on my 'average' day, if such a thing exists. Currently I am supporting three client companies in the metropolitan area of Harrisburg, Pennsylvania. The largest is a multi-national corporation currently employing approximately 40,000 people. The smallest is a 'Mom and Pop' PC store, which employs ten (mostly sales staff). Each client is different, requiring varying degrees of administrative detail, on-site attention and level of support. However, since all my clients require the same thing in the end (protection, advance warning, and occasionally reaction), it is relatively easy to support them all.

The Day Starts

The alarm clock goes off at 5am. After dropping the little one off at Grandma's (low cost childcare, what can I say?), it is off to do some on-site care. Reaching my office, housed in a corporate systems security area, I brew a quick coffee while the systems start up. Once operational, last night's incident logs can be reviewed and it is not unusual for this company to see 20 to 30 incidents in a 24-hour period. Unfortunately, I know the logs are under-reported.

During this time, I consolidate the names of users and the viruses encountered. A trouble ticket is generated for any in-house employee, and a local LAN Support person is dispatched to ensure the victim's PC has the current corporate-approved anti-virus product installed and up to date. Any viruses not previously encountered are sampled and replicated for inclusion in the monthly report for the WildList. After the logs are processed and the required charts and reports are generated, it is time to review email, which usually consists of the morning dose of spam, user questions, hoaxes, correspondence from colleagues and the occasional LAN Supporter submitting suspicious files.

While a special mailbox and procedures have been established for verifying alerts and suspect files, it is not always easy to get everyone to comply. All attachments, even the ones that are memos or not being submitted as a sample are saved and examined on a goat machine. Any attachment found to be viral is reported back to the submitting user. If the current corporate-approved product, with up-to-date signatures, detects the virus, the submitter is notified to update. If, on the other hand, the product does not detect it, more steps are required.

The file is then checked against other products. If any of them detect the virus, it is replicated to a goat file to preserve company confidentiality, and submitted to the primary vendor with the name provided by the backup product (as well as the backup product's name). If none of the products on hand detect the virus, the file is examined and if appropriate, replicated onto goat files and submitted to the vendor. In both cases, the virus is marked and not allowed for inclusion with the report for the WildList since the vendor will report it and that would artificially move the virus to wild status.

Once the suspect file has been confirmed as viral, and named by the vendor, an alert is posted on an Intranet Alert Page, and email (with cure, when possible) is sent to LAN Support personnel. In case anyone is wondering, my email is replicated to all four of my email accounts, so I do not miss anything, regardless of my location.

Taking advantage of the T1 Internet link, the next step is to review vendor Web pages. While I subscribe to as many 'Alert' mailing lists as I can, there are times when a new virus alert is posted to a Web site before the email gets to me. Reviewing the sites also gives me an insight into what vendors are planning for the future, as well as knowing what their current revisions, patches and signatures are.

If there are any software updates, I retrieve them. I also use this time to read through alt.comp.virus, as there are the occasional 'gold nuggets' that make handling new threats easier. I try hard to give unbiased assistance, through private email, to as many users as possible – a little free assistance may lead to a contract. If nothing of significance has occurred, or is occurring, I like to check in on the *CompuServe* forums. These days, *CompuServe* is largely neglected by many anti-virus vendors – what were once thriving communities are down to a handful of diehard regulars supporting users with problems.

Phone Support

Since time is money, I try to maximize my time by using email and the telephone to do the majority of my support. It would not be practical for me to pack up and travel two

states, or even to another country, for a simple configuration issue. While vendors do provide their customers with support options, my customers like my personal, hands-on touch and often come to me for first resolution. I feel that since I designed their strategy, I should support the product they choose, even though I may not be a product specialist. The only way for all these factors to work together is to have a good solid working relationship with all facets of the Network Support teams.

Since LAN Administrators are my eyes and ears to the problem, it is key to communicate with them. The problems phoned in from LAN Administrators tend to run the gamut from 'Your product is crashing my server' through 'What is the best configuration to use in my environment?' to 'I think I have a new virus here'. Of course, all calls are logged as to length and detail of resolution. When doing phone support, it is important not to act like you are rushing them off the phone, but at the same time to keep the call as short as possible.

Hoax and Non-viral Support

Like everyone else who deals with computer viruses, I have to deal with the inevitable 'Is this a virus?' questions. The number of hoax messages dealt with this year are way down on the previous year's, but I still spend too much time debunking them. My most successful campaign for dealing with virus hoaxes involved setting up a Web page with five heuristics for detecting a hoax, and providing a link to Rob Rosenberger's Virus Myths page.

To supplement this, a policy was made prohibiting anyone outside computer security from initiating a 'virus alert'. This policy requires all users to submit any security alerts of any type to the computer security department for verification, and then, if the situation warrants, they release an alert via email and a special Web page. This policy is reiterated every six months by administrative email and voicemail. While it has not eliminated hoaxes, it has put a great dent into them.

On an average of at least three times a week I receive a call or email where the user has received an unsolicited email and thinks it is a virus. This has become more prevalent since Melissa and ExploreZip. In some of these cases, the email has neither attachments nor viral symptoms other than it appeared in my mailbox. As you can guess, a good number of these are spams for anything and everything. Occasionally they are jokes and such from friends with an unexpected email address. While malware like Melissa and ExploreZip have made many users more aware of the potential problems, this has had the side effect of making many people paranoid.

Emergencies

Like everyone else, we have faced several viral 'emergencies' with viruses that move so fast, or so far, that it threatens the company's network integrity. This can be anything from a new Laroux variant to an ExploreZip-type situation. It is during these incidents that you really earn your keep and when it is critical to have some type of contact with one or more vendor's technical group. For the more mundane of these incidents it is a matter of isolating the sample and getting the vendor to issue a new set of signatures. Usually, this is easy – the hard part is deploying this cure to an environment that spans over seven languages and 53 countries. Once again, local administrators are critical in the implementation.

Unusual situations include the new, fast, network infectors like Melissa and ExploreZip. Industry contacts, other corporate researchers, vendor employees etc are very helpful in these situations. My first indicators of both these threats were either phone calls or email which simply stated, 'Have you seen this? If not, you will, it's all over the place'. With the initial warning, I occasionally get samples. By analysing (both disassembling and secure test infection observation), I coordinate a defensive effort with all my clients. Of course we wait for an official 'fix' detector from the vendor, but we also create proactive measures.

Since both Melissa and ExploreZip were 'fixed targets' (some part of the incoming email file stayed the same), we were able to create some rules and tools which temporarily blocked these messages. While recognized as a short-term fix, the measures were sufficient to protect until the anti-virus vendor could provide updates. For the curious, several instances were incurred involving both viruses, with no penetration while these crude tools were in use.

Still To Do

I return home with anti-virus reconnaissance still to perform. To protect my clients better, I visit several VX sites, to see what they have to download, what they are saying in their newsletters and the like. This indicates what they are capable of and what they are planning. While I have an ethical distaste for downloading viruses from these sites, their newsletters often prove valuable. Once this is done, I retire off to the Lab and test any new product versions. A nominal test is performed which only checks new features and enhancements since the last formal product review, carried out only by special request.

Conclusion

Normally I do not write cures for new viruses – I leave that to the vendors – but I find that I am doing much the same as my colleagues employed by anti-virus companies. Clients depend on individuals like me to be the intermediary between them and the anti-virus vendor. When a virus gets through the defensive net, it is up to us to halt it in its tracks. I do not consider myself an expert but it is often my knowledge and forethought that keeps an organization from losing more money than it would have if it did not implement a full anti-virus procedure. I hope people realize after reading this that the independent corporate reporters are where the rubber meets the road.

FEATURE SERIES

Macro Viruses – Part 1

Dr Igor Muttik
AVERT Labs, UK

Macro viruses appeared about four years ago and are now the most prevalent in the field. Their number is growing very quickly (currently about 5,000). The macro virus category is developing swiftly and many new terms and notions are invented constantly, so it might be difficult to keep up to date with them. It is easy to get lost in words like 'mating', 'remnants', 'downconversion' until you know what they actually mean.

This series gives an insight into the environment in which macro viruses live (OLE2 files), summarizes the main features of macro viruses and of their host applications, explains currently used terminology and provides a basic knowledge of how macro viruses operate.

What is a Macro?

Many applications provide the functionality to create macros. A macro is a series of commands to perform some application-specific task. Macros are designed to make life easier, for example by performing everyday tasks like text formatting or calculations in spreadsheets.

Macros can be saved as a series of keystrokes (the application records which keys you press). They can also be written in special macro languages (usually based on real programming languages like C and BASIC). Modern applications combine both approaches and their advanced macro languages are as complex as general purpose programming languages. When the language allows the modification of files it becomes possible to create macros that copy themselves from one file to another. Such self-replicating macros are called macro viruses.

A Brief History

Many software packages have a macro language – perhaps the very first well-known and widespread one was the *Lotus 123* spreadsheet. It was proved long ago that it is possible for *Lotus 123* to write a self-replicating macro (a virus) which will be able to travel from one file to another. However, viruses have never been a problem for *Lotus 123* as its macro language is rather simple and access to files can only be performed via menus. So, a virus for *Lotus 123* would be extremely obvious – you would literally see the infection process right on your screen.

In December 1994, the researcher Joel McNamara wrote the first real macro virus for demonstration purposes. It was called DMV (Document Macro Virus). In fact, there were two viruses written – DMV for *WinWord* and DMV for *Excel*. The samples were used to demonstrate the possibility of macro viruses on these platforms. The first field macro virus – WM/Concept – appeared in the summer of 1995 and soon became the most widespread virus ever.

Platforms and Applications Supporting Macros

Most macro viruses are written for *Microsoft WinWord* and *Excel*. Viruses for *PowerPoint 97* also exist, even in the wild (PP97M/Tristate). However, there are also experimental macro viruses for *AmiPro* (Green_Stripe), *CorelDRAW* (CSC/CSV, etc.), *Access 97* (AccessiV, etc.) and several multi-partite viruses which infect executable files and *WinWord* documents (Anarchy.6093, Heathen).

Macro viruses can work on any machine carrying, say, *WinWord* – be it a PC, a Macintosh or a DEC Alpha computer. Macro hosting applications are able to work under many operating systems – *Windows 3*, *Windows 95*, *Windows NT*, MacOS, *SoftWindows*, etc. There are certain differences in implementations of the macro languages on different machines (OS support is usually slightly different, especially for the filesystem objects) but nevertheless, many macro viruses can spread successfully on very different types of computers and operating systems.

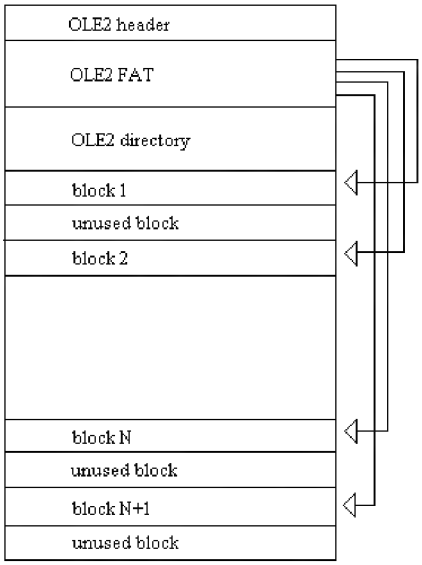
OLE2

Files produced by *Microsoft* applications (DOC documents, XLS spreadsheets, PPT presentations created by all versions of *WinWord* above 6.0, and all versions of *Excel* and *PowerPoint*) are stored in so-called OLE2 files (note, *MS-Access* files are not OLE2). OLE stands for Object Linking and Embedding. It is just a standard describing a file structure that is able to store many different streams within one file. An OLE2 file is a file system within a file.

OLE2 files contain a special signature at the beginning (D0 CF 11 E0 – which stands for DOCFILE), the FAT (File Allocation Table), and a directory just like a normal DOS disk. Space inside an OLE2 file is allocated in blocks referenced from the OLE2 FAT.

The access to OLE2 files is supposed to be gained through APIs provided by OLE2.DLL and OLE32.DLL. These DLL files support all necessary functionality to work with OLE2 files (like add/delete/modify stream, open/update an OLE2 file, etc.) The OLE2 technology is being licensed to other software producers, so many vendors are now supporting this format.

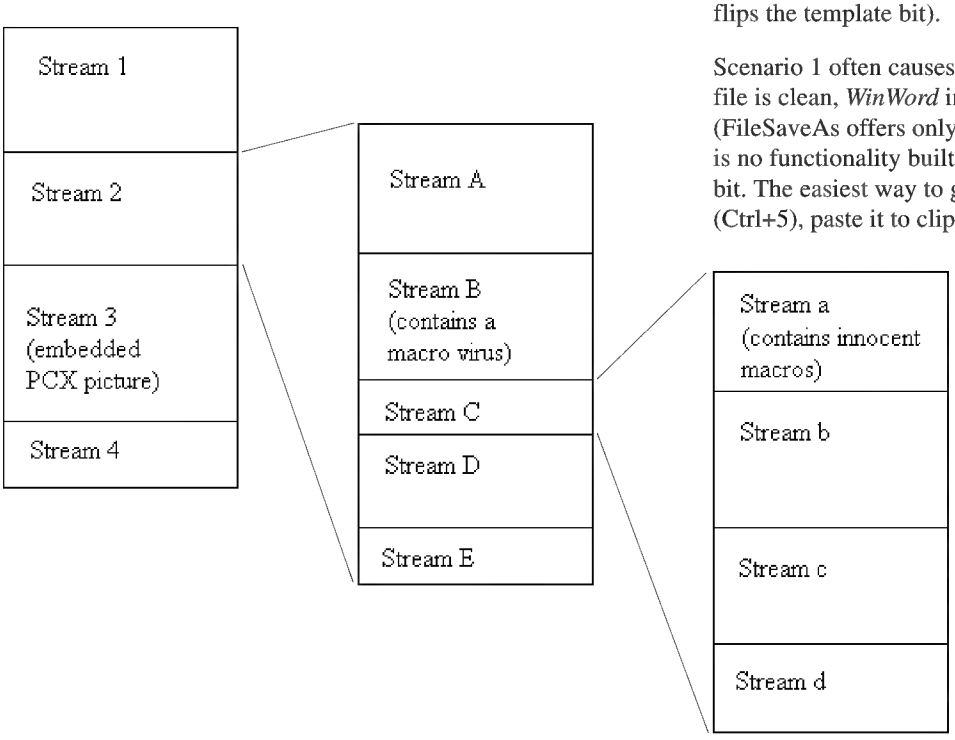
The flexibility of the OLE2 format allows the storage of many not necessarily related items (they are called streams) inside just one file. For example, the first stream of an OLE2 file may hold the text, the second another OLE2 file, the third an embedded picture, etc, see diagram.



As you can see, OLE2 files may have unused blocks. They usually contain information from previous saves of the file or just some random data from memory (even the important private data you may not want to be included in any file!).

A very common situation is when, say, an XLS file has a lot of strings reading 'laroux' inside but they are all in the unused space. This situation frequently causes users to panic while the file appears to be clean with no macros, let alone infected by the Laroux virus. To avoid files having unused clusters uncheck 'Allow fast saves' in Tools/Options/Save.

In the following example the first OLE2 file has an embedded OLE2 file as Stream 2 and a PCX file as Stream 3. The embedded OLE2 storage contains a virus in Stream B and another embedded OLE2 storage as Stream C. This third OLE2 contains some macros in Stream a. This multi-level structure is all held within one real file which is organized as a file system on each level (because embedded objects are also in OLE2 format).



In an OLE2 file it is possible to embed an XLS into a PPT file (or an EXE into DOC, or a DOC into PPT). In this way, tree-like structures within OLE2 may be created. If the embedded object is being double-clicked on, its contents are activated and macros (if any) may be executed.

A further complication is that to save space, *PowerPoint* stores embedded OLE2 files in compressed form. So a PPT file is an ordinary OLE2 file but it can have another compressed OLE2 embedded. To be able to scan for macro viruses inside the OLE2 files on all levels of embedding scanners usually use their own OLE2 parsing and decompressing engine. That allows the scanning of *WinWord* documents directly, even without support of *Windows'* OLE2.DLL and OLE32.DLL. Decent scanners are able to scan OLE2 files even under DOS and *NT* on a *Novell* server, a Unix machine, Macintosh, DEC Alpha, Sun, etc.

Template bit, DOC/DOT

WinWord 6/7 documents have a special bit inside which says whether the current document contains anything but text. *WinWord 6/7* does not look for macros if the template bit is zero. Normally, DOC files have this bit reset (zero) and templates (DOT) set to 1. However, the bit itself is not linked to the file extension (and on Macintosh there are no fixed extensions for files).

So, it is possible to have: 1 – a file with no macros and the template bit set (this normally does not happen but can happen when all macros are removed from the DOC file), 2 – a file with macros (e.g. virus) and the template bit set (this is a normally infected file), or 3 – a file with macros (e.g. virus) and the template bit reset (this means the virus is inactive, or 'dormant' – it will not infect until somebody flips the template bit).

Scenario 1 often causes the user confusion. Even when a file is clean, *WinWord* insists on saving it as a template (FileSaveAs offers only 'Document Template' type). There is no functionality built into *WinWord* to clear the template bit. The easiest way to get rid of it is to select whole text (Ctrl+5), paste it to clipboard (Ctrl+C), close the file (Ctrl+W), start new file (File/New) and paste the text back (Ctrl+V). Now FileSaveAs will work fine.

Office 97 and *Office 2000* ignore the template bit and check for the presence of macro storage. However, it is possible to have an *Office 97* file with empty storage (i.e. no macros). Then an *Office* application would display the macro warning box even for a file with no macros whatsoever.

[*Next month, the second instalment covers WordBasic,VBA, up/down conversion and polymorphism. Ed.*]

COMPARATIVE REVIEW

Any ImprovemeNT?

Six months, an English summer and much Internet Worm excitement have passed since the last *NT* comparative, back in March of this year. Then it was eighteen products that were submitted for review. Sixteen are present this time.

Test Procedures

As usual for the *VB* comparatives, three essentially identical test machines were used for the product testing. The hard drives of each were completely wiped with a fresh *NT 4.0* (SP 5) image prior to the testing of each product. To eliminate any potential discrepancies, all speed tests (scan rates and scanner overheads) were performed upon one of the machines, whilst disconnected from any network.

The test-sets were updated from those used in the previous comparative, and importantly, the In the Wild (ItW) File and Boot sets were aligned to the June 1999 WildList. New additions to the ItW viruses included W97M/Pri.A, W97M/Walker.E, W97M/Walker.F, and the email propagating Win32/ExploreZip and Win32/PrettyPark Worms. The COM/EXE infecting ACG.B joins ACG.A in the Polymorphic set, and the Macro set welcomes W97M/ZMK.P, the B, C and D variants of W97M/Lys, and W97M/Melissa.I amongst others. Additionally, samples infected with {W95,W97M}/Heathen.A, a virus capable of infecting both *Windows* executables and *Word* documents, have been added to the Standard and Macro test-sets. For a complete listing of the viruses in each of the test-sets, see the URL quoted at the end of this review.

Speed tests were performed in order to assess two aspects of each of the products. Firstly, the overhead of each of the on-access scanners was assessed, by measuring the time taken to copy a set of 100 executable and 100 OLE2 files between directories, with the on-access scanner in a variety of configurations. For presentation in this review, the results have been normalized with respect to a common baseline of 17 seconds, enabling them to be presented in units of time. Next, the scanning speed of the on-demand scanners were measured, by timing how long it took to scan a set of 5,500 COM and EXE executables (520 MB), and a set of 373 OLE2 files (65.3 MB). These latter tests double up as false positive tests, since all the files are clean and no viruses should be detected.

On-demand tests were performed whilst logged in as Administrator on the workstation. The test-sets were stored on a network drive as a read-only share. For products that were incapable of scanning network drives, the test-set was copied to a local hard drive. On-access detection rates were determined with the usual *VB* method – using a utility that recursively searches the test-set directory tree, attempting to open each of the files encountered. For scanners where the option to 'deny access' to suspected files was unavailable, the configuration was altered to scan on file writes, and delete infected files. Subsequently, the test-set was copied to a local hard drive. In some cases it was necessary to copy the test-set repeatedly between different directories on the hard drive until no further infections were found. This latter testing method was also applicable to products that could only scan on file writes.

Full details of the results are presented in the main tables. The brief results summary presented under each of the product headings are those for on-demand scanning unless otherwise indicated.

Alwil Avast32 v3.0-154 (24/6/99)

ItW Overall	99.7%	Macro	95.3%
ItW Overall (o/a)	98.2%	Standard	98.4%
ItW File	99.7%	Polymorphic	93.9%

Since its last appearance, *Avast32* has received a fair amount of attention from its developers at *Alwil*. As with many of the other products, files of *PowerPoint* format are now supported, as is scanning within ZIP archives.

On-demand detection rates are respectably high – only the failure to detect one of the three Win95/Kenston samples prevented *Avast32* from claiming the VB 100% award. A variety of samples were missed from the other test-sets – a handful of Marburg-infected executables, the polymorphic X97M/Soldier.A, and the {W32, W97M}/Heathen.A samples, a recent addition to the test-set.

VB has been unable to test the on-access scanner of *Avast32* in previous tests, due to its dependence upon file execution. This latest version scans on file *writes* however, and so for the first time, the standard of *Avast32's* real-time protection has been assessed. Detection rates were determined by copying the test-set to the local hard drive with the scanner set to delete infected files. The copied files were then copied between directories on the local hard drive, until after three iterations of the process, no further infections were found. On the whole, detection rates were lower than those observed during on-demand scanning.

Testing on-access scanning of the ItW boot viruses proved wearing. As with other products in this and previous comparatives, *Avast32* failed to detect disk changes reliably. Detection (or not) also seemed to depend upon the sequence in which the test disks were checked. Admittedly, bombarding a scanner with a large number of diskettes infected with different boot viruses may not be a *realistic* scenario, but these observations do reveal a slight weakness in the on-access scanner's architecture.

On-demand tests	ItW Boot		ItW File		ItW Overall	Macro		Polymorphic		Standard	
	Missed	%	Missed	%	%	Missed	%	Missed	%	Missed	%
Alwil Avast32	0	100.0%	1	99.7%	99.7%	142	95.3%	273	93.9%	22	98.4%
CA InnoculateIT	0	100.0%	0	100.0%	100.0%	7	99.7%	174	96.9%	1	99.9%
CA Vet Anti-Virus	0	100.0%	0	100.0%	100.0%	22	99.4%	268	93.9%	3	99.7%
Command AntiVirus	0	100.0%	2	99.4%	99.4%	14	99.8%	112	98.0%	0	100.0%
Data Fellows FSAV	0	100.0%	4	99.7%	99.7%	20	99.4%	16	99.7%	0	100.0%
Dialogue Science DrWeb32	0	100.0%	2	99.1%	99.1%	18	99.3%	10	99.8%	1	99.7%
Eset NOD32	0	100.0%	0	100.0%	100.0%	7	99.7%	0	100.0%	1	99.7%
GeCAD RAV	0	100.0%	0	100.0%	100.0%	25	99.1%	503	96.9%	82	94.3%
Grisoft AVG	0	100.0%	3	99.1%	99.1%	55	98.3%	96	96.8%	32	98.6%
Kaspersky Lab AVP	0	100.0%	0	100.0%	100.0%	0	100.0%	0	100.0%	0	100.0%
NAI VirusScan	0	100.0%	1	99.9%	99.9%	3	99.9%	0	100.0%	0	100.0%
Norman Virus Control	0	100.0%	0	100.0%	100.0%	5	99.8%	174	96.9%	0	100.0%
Proland Protector Plus	3	91.8%	81	89.2%	89.4%	1104	62.8%	11138	22.1%	515	65.2%
Sophos Anti-Virus	0	100.0%	9	97.9%	98.1%	53	98.2%	174	96.9%	12	99.5%
Stiller Integrity Master	0	100.0%	201	64.5%	66.7%	1555	50.2%	10143	29.8%	255	83.9%
Symantec Norton AntiVirus	1	97.3%	0	100.0%	99.8%	14	99.4%	264	93.9%	1	99.7%

CA InnoculateIT v4.53 (24/6/99)

ItW Overall	100.0%	Macro	99.7%
ItW Overall (o/a)	98.6%	Standard	99.9%
ItW File	100.0%	Polymorphic	96.9%

Now the proud owners of *Cybec's Vet Anti-Virus*, it will be interesting to monitor how *Computer Associates* develops its two anti-virus siblings. Despite being obviously different products, confusion between the two will almost certainly exist, especially since the *Innoculate IT Personal Edition* that is available for free download from the *CA* site, is in fact, the *Vet* product in disguise. The product reviewed here is the *Enterprise Edition*, that native to *CA*.

InnoculateIT, has put in some solid performances over recent comparatives – its only downfall has been its stability. Thankfully, during testing of this version of the product no serious stability problems were encountered. However, testing the overhead of the on-access scanner proved problematic when it was set to scan incoming files. The usual *VB* method of measuring overhead was employed, which, for most products, returns very similar times for each iteration of the copying process. With *InnoculateIT* however, the times were extremely erratic, and it was not possible to obtain a consistent set of times. The results quoted are therefore an average of all the times recorded.

Detection-wise, the product maintains the high standards it has set previously, attaining the VB 100% award again. Results were poorer across all the test-sets during on-access scanning, due partly to the failure to check sufficient file types. This was most in evidence in the ItW and Polymorphic sets, where screen saver (SCR) samples infected with Marburg and TPVO.3783.A slipped through the net.

CA Vet Anti-Virus v10.0.2 (2/7/99)

ItW Overall	100.0%	Macro	99.4%
ItW Overall (o/a)	99.8%	Standard	99.7%
ItW File	100.0%	Polymorphic	93.9%

When commencing the testing of some of the products submitted to *VB*, there is often a feeling of apprehension, as a multitude of potential problems are anticipated. Not so, with

Vet Anti-Virus. *Vet* has always been second to none in terms of stability, and, clearly, its detection capabilities are equally competitive. It was in September 1998 that *Vet* last earned the VB 100% award, and so perhaps it is fitting that a year on, it achieves that status again.

Interestingly, four Marburg samples were missed from the Polymorphic test-set during both on-demand and on-access scanning. As with the majority of the products, detection rates were generally lower during on-access scanning, where, in this case, O97M/Tristate.C infected *PowerPoint* samples were missed from the ItW set.

Command AntiVirus v4.57β (1/7/99)

ItW Overall	99.4%	Macro	99.8%
ItW Overall (o/a)	98.8%	Standard	100.0%
ItW File	99.4%	Polymorphic	98.0%

Failure to detect two of the three samples of Pieck.4444.A in the ItW set kept the VB 100% award at arm's length from *Command Software AntiVirus* (*CSAV*).

Elsewhere, detection rates were high. In the Polymorphic set, the bulk of the misses were due to only a third of the ACG.A samples being detected. In the Macro set, all the samples infected with the polymorphic X97M/Soldier were missed, as was one of the three PP97M/Vic.A samples.

CSAV's performance in the speed tests was fairly average, giving a throughput of approximately 1400 and 2300 KB/s for scanning executable and OLE2 files respectively. The *Dynamic Virus Protection* (*DVP*) facility that is the on-access scanner of *CSAV* induced a reasonably large overhead of just over 220% when enabled.

Data Fellows F-Secure Anti-Virus v4.04

ItW Overall	99.7%	Macro	99.4%
ItW Overall (o/a)	99.2%	Standard	100.0%
ItW File	99.7%	Polymorphic	99.7%

Data Fellows F-Secure Anti-Virus (*FSAV*) keeps up the high standard of detection set by the other products so far in this review. Failure to cope successfully with *PowerPoint* file formats resulted in missing all the samples infected with the A, B, C and D variants of O97M/Tristate, PP97M/Vic.A and PP97M/Shaper.A, for both on-demand and on-access scanning. A handful of ACG.A samples was also missed from the Polymorphic set.

Results were slightly poorer for on-access scanning, due mainly to missing the VxD samples of Win95/Fono, Win95/Navrhar and Win32/PrettyPark.

As ever, the use of two detection engines in the one product gives the expected results – high detection rates but only a mediocre scanning speed. This was also reflected in the overhead of the real-time monitor, *GateKeeper*, which at 225% was slightly above the average observed across the products. One or other of the detection engines could be removed from *FSAV*, although whether such a sacrifice to the detection capabilities would be worth it in terms of scanning speed is doubtful.

On-access tests	ItW Boot		ItW File		ItW Overall	Macro		Polymorphic		Standard	
	Number	%	Number	%	%	Number	%	Number	%	Number	%
Alwil Avast32	3	91.8%	7	98.6%	98.2%	146	95.3%	311	92.9%	8	99.5%
CA InnoculateIT	3	91.8%	16	99.0%	98.6%	36	98.9%	420	95.9%	1	99.9%
CA Vet Anti-Virus	0	100.0%	3	99.7%	99.8%	38	98.9%	768	90.8%	6	99.5%
Command AntiVirus	3	91.8%	3	99.3%	98.8%	17	99.7%	112	98.0%	0	100.0%
Data Fellows FSAV	0	100.0%	6	99.1%	99.2%	28	99.1%	23	99.6%	9	99.7%
Eset NOD32	0	100.0%	0	100.0%	100.0%	7	99.7%	2	99.9%	1	99.7%
GeCAD RAV	n/a	n/a	0	100.0%	n/a	13	99.5%	503	96.9%	82	94.3%
Grisoft AVG	0	100.0%	4	99.0%	99.1%	61	98.2%	268	93.9%	112	91.6%
Kaspersky Lab AVP	0	100.0%	0	100.0%	100.0%	0	100.0%	0	100.0%	0	100.0%
NAI VirusScan	0	100.0%	1	99.9%	99.9%	3	99.9%	0	100.0%	0	100.0%
Norman Virus Control	3	91.8%	7	99.4%	99.0%	50	98.6%	177	96.7%	0	100.0%
Sophos Anti-Virus	0	100.0%	8	98.0%	98.1%	52	98.2%	174	96.9%	12	99.5%
Symantec Norton AntiVirus	1	97.3%	0	100.0%	99.8%	14	99.4%	264	93.9%	1	99.7%

Dialogue Science DrWeb32 v4.11 (2/7/99)

ItW Overall	99.1%	Macro	99.3%
ItW Overall (o/a)	n/a	Standard	99.7%
ItW File	99.1%	Polymorphic	99.8%

Processing the variety of log files produced by sixteen different products is a task enough by itself. Generally, problems exist with products that use multiple tags within the same log to mark infected files. However, *DrWeb32* introduced a new dimension to the task by logging certain scanned files as both clean and infected!

This slight inconvenience aside, *DrWeb32* achieved high detection rates across all the test-sets, although failing to detect Win95/PrettyPark cost the Russian product the VB 100% award.

Currently *DrWeb32* does not incorporate an on-access scanner, an issue which is currently being addressed by the developers. Come the next comparative, when on-access scanning is incorporated into the VB 100% award, the performance of this component will be of much interest.

Eset NOD32 v1.20 (2/7/99)

ItW Overall	100.0%	Macro	99.7%
ItW Overall (o/a)	100.0%	Standard	99.7%
ItW File	100.0%	Polymorphic	100.0%

Another product proving straightforward to test was this Slovak offering. An anti-virus product in the strictest sense of the term, not jam-packed with additional features, *NOD32* does what it claims extremely well. Only eight and ten samples were missed across all the test-sets during on-demand and on-access scanning respectively.

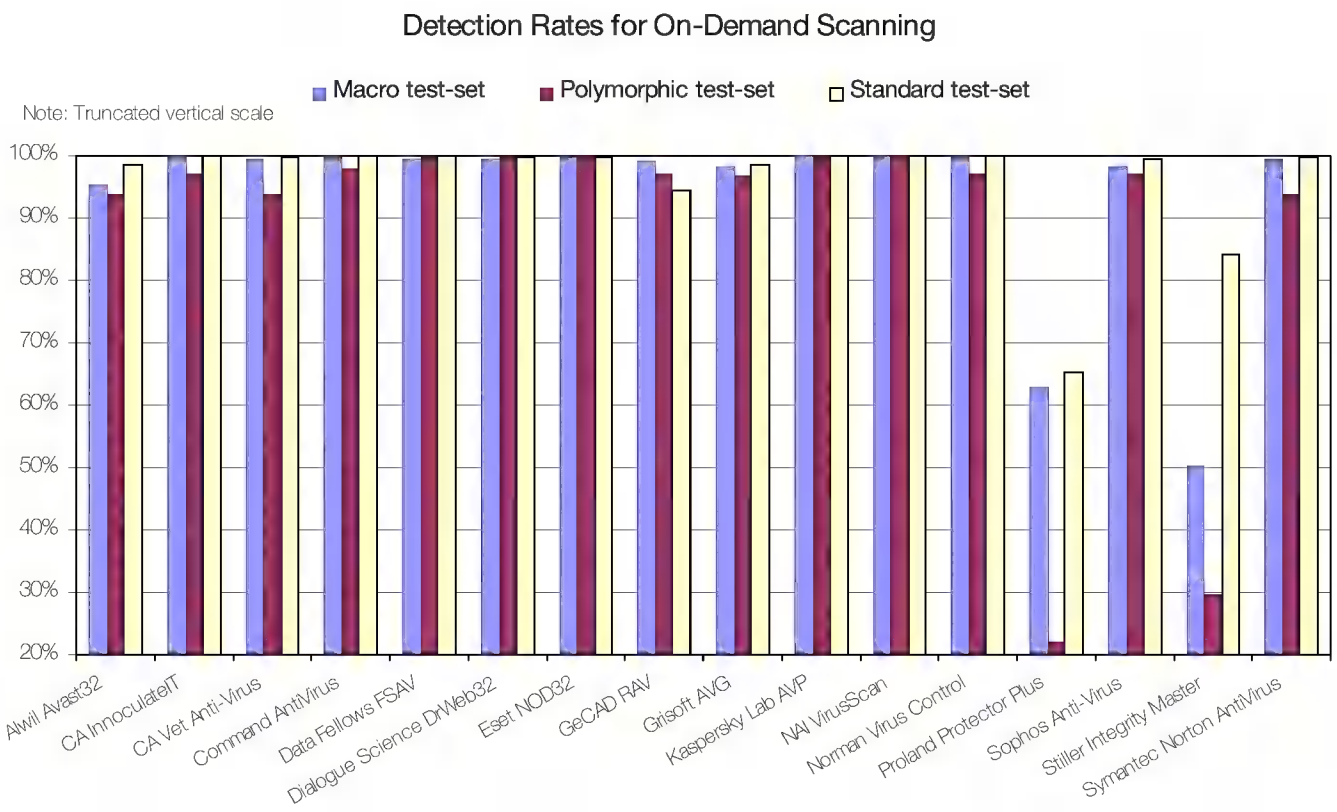
These misses were registered against samples infected with {Win95/W97M}/Heathen.A, and document templates infected with the B, C and D variants of W97M/Lys. Two samples of the polymorphic Nightfall.4518.B were also missed by the on-access scanner.

GeCAD RAV v7.0 (2/7/99)

ItW Overall	100.0%	Macro	99.1%
ItW Overall (o/a)	n/a	Standard	94.3%
ItW File	100.0%	Polymorphic	96.9%

The recipient of a major facelift quite recently, *RAV 7* is the first of *GeCAD's Romanian Anti-Virus* products to sport an on-access scanner. Such a feature is pretty much essential for any product vying for attention in the anti-virus arena today.

Achieving complete detection of the ItW file collection in both on-demand and on-access scanning will certainly please the developers. Unfortunately, on-access scanning of



floppy boot sectors was not supported in the submitted version of *RAV*, and so overall ItW detection of the on-access scanner can not be assessed. *VB* has been informed that plans are currently afoot to address this deficiency.

Outside of the ItW sets, detection rates were not as high as some of the other products. Failing to detect all the samples of Neuroquila.A, and a few of the ACG.A samples accounted for the misses amongst the Polymorphic test-set, and a variety of misses were registered in the Standard set.

Grisoft AVG v6.0 (28/6/99)

ItW Overall	99.1%	Macro	98.3%
ItW Overall (o/a)	99.1%	Standard	98.6%
ItW File	99.1%	Polymorphic	96.8%

Grisoft's AVG is yet another product to have benefitted from a recent makeover, and also features an on-access scanner for the first time in a *VB* review. The slightly unusual interface still forms the main operations centre, although improvements have been made to ease the task of configuration alteration.

Overall, this was a strong showing from *AVG* – detection rates were respectably high across all the test-sets. Previous problems that have been encountered with detection of infected floppy boot sectors with invalid BPB's appear to have been fixed, and all the ItW boot viruses were detected, both on-demand and on-access. Unfortunately however, Win95/Padania was missed in the ItW file set, keeping the VB 100% award at bay.

In terms of speed, *AVG* is at the lower end of the products tested for scanning executables, although far speedier when it comes to OLE2 files. Unfortunately however, the in-built heuristics which are responsible for a good proportion of the correct detections in the above tests, led to unwelcome false positives in the speed tests.

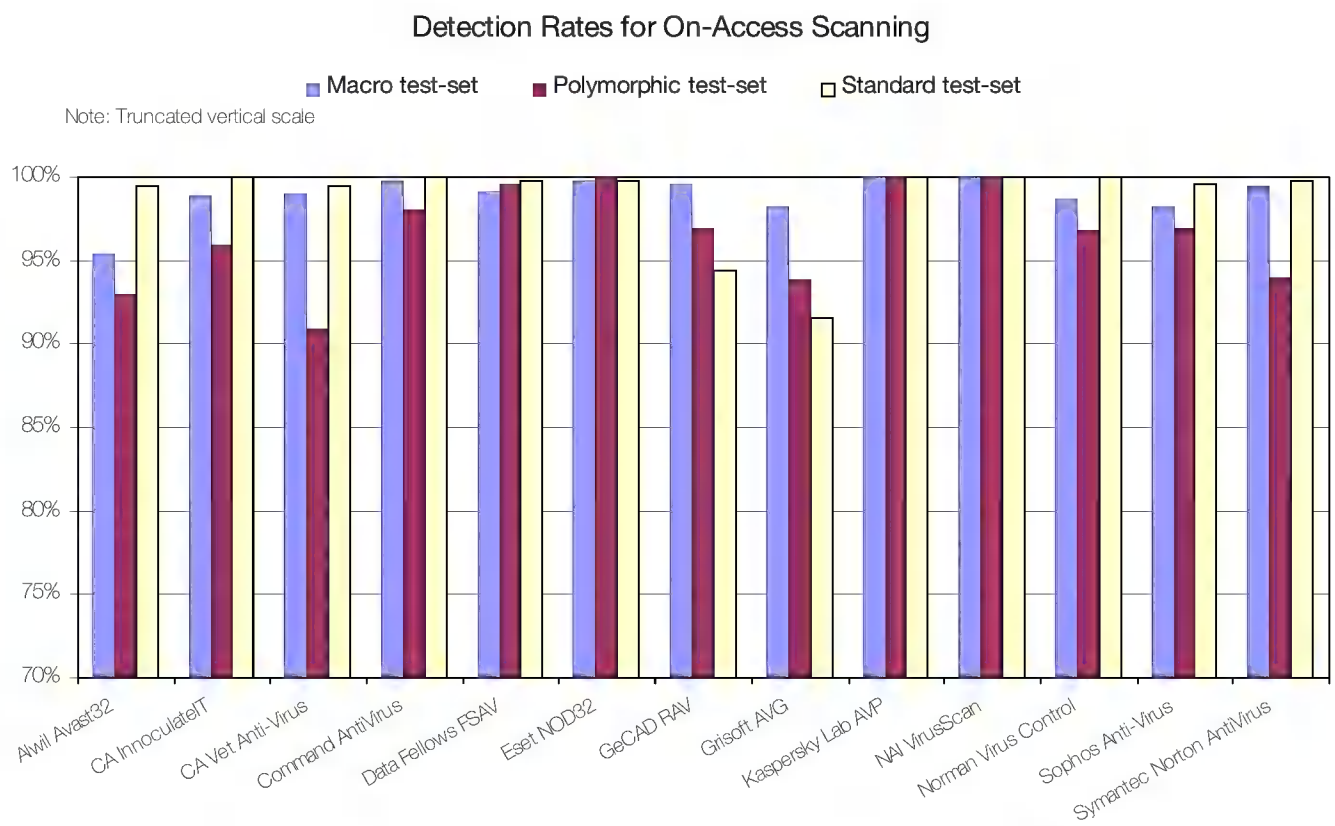
Kaspersky Lab AVP v3.0.131 (30/6/99)

ItW Overall	100.0%	Macro	100.0%
ItW Overall (o/a)	100.0%	Standard	100.0%
ItW File	100.0%	Polymorphic	100.0%

Long the recipient of praise for achieving high detection rates, *Kaspersky Lab's AVP* detected all of the samples during on-demand scanning this time around, and is thus the fifth claimant to the VB 100% award. After such an impressive start, the strength of this Russian product was driven home further when the achievement was repeated by the on-access scanner. Impressive indeed.

The only blemish on the product occurred during the speed tests where two executable files were falsely identified as suspicious, and a third was declared to be corrupted. As noted for *AVG*, this is the negative effect of the heuristics which help to boost the detection rates. *AVP* is in the upper half of the field in terms of scanning speed.

The on-access scanner, giving an overhead of just over 200%, imposes itself a little more than some of the other scanners, but was far from the worst in this respect.



NAI VirusScan NT v4.03a.4032 (30/6/99)

ItW Overall	99.9%	Macro	99.9%
ItW Overall (o/a)	99.9%	Standard	100.0%
ItW File	99.9%	Polymorphic	100.0%

Though not a clean sweep as for its alphabetical predecessor, *Network Associate's VirusScan NT* is following hot in *AVP's* footsteps.

This is due partly to the fact that the default file extension list has finally been updated such that file types associated with viruses known to be in-the-wild are scanned by default. Only the extensionless samples of the A, B, C and D variants of O97M/Tristate were missed throughout the test-sets in both on-demand and on-access scanning.

VirusScan is in the middle of the pack when it comes to scanning speed and on-access scanner overhead. Pleasingly, no false positives were detected during the speed tests.

Norman Virus Control v4.70 (1/7/99)

ItW Overall	100.0%	Macro	99.8%
ItW Overall (o/a)	99.0%	Standard	100.0%
ItW File	100.0%	Polymorphic	96.9%

Recently featured in a standalone review (see *VB*, August 1999, p.21) *Norman Virus Control* (*NVC*) is the final product of this comparative to detect all the ItW File and Boot viruses during on-demand scanning, and thus achieve the VB 100% award.

On-demand scanning was reasonably quick, and three viruses account for all the misses that were observed – ACG.A, W97M/Stat.A and a document template infected with WM/Triple.B. Results were not so promising during on-access scanning however, exposing a slight weakness in *NVC*. Three ItW boot virus samples were missed (those with invalid BPB's), as were samples infected with XM/Compat.A and O97M/Tristate.C in the ItW file set.

Proland Protector Plus v6.6

ItW Overall	89.4%	Macro	62.8%
ItW Overall (o/a)	n/a	Standard	65.2%
ItW File	89.2%	Polymorphic	22.1%

In the last *VB* review of *Proland's NT* offering, it was mentioned that the product had some 'maturing' to do. Well, six months have passed by since then, and a greater degree of maturity is certainly evident in the results presented this time around.

An awful lot of samples were still missed from the Macro, Standard and Polymorphic test-sets however, especially the latter. The results clearly indicate that *Proland's* developers have focused predominantly upon ItW virus detection.

Whilst performing the speed tests, it was not possible to complete a scan of the OLE2 file set, due to a recurring application error. Consequently scanning speed results are limited to the scanning of executables. The decrease in scanning speed compared to that observed previously is concurrent with the general increase in the detection rates.

Sophos Anti-Virus v3.23

ItW Overall	98.1%	Macro	98.2%
ItW Overall (o/a)	98.1%	Standard	99.5%
ItW File	97.9%	Polymorphic	96.9%

Failure to detect samples infected with Win95/Padania and O97M/Tristate.C prevented *Sophos Anti-Virus* (*SAV*) from achieving complete detection of the ItW viruses.

From the log files produced during on-demand scanning, a slight oddity with the treatment of the extensionless 'Book1' samples infected with O97M/Tristate was noticed. Notably, some were scanned and successfully detected, despite the scanner configuration supposedly excluding files with no extension. It transpired that a minor bug in the product (no longer present in the current product) caused such files to be scanned – a 'positive' bug in this case!

In terms of stability *SAV* proved to be one of the top products again, reliably detecting all the boot sector viruses in both on-demand and on-access scanning. It was also one of the few products whose on-access scanner was up to the standard of the on-demand scanner. This will, no doubt, stand it in good stead when on-access scanning is introduced into the VB 100% awards, as of the next comparative in the November issue.

Stiller Integrity Master v4.21a

ItW Overall	66.7%	Macro	50.2%
ItW Overall (o/a)	n/a	Standard	83.9%
ItW File	64.5%	Polymorphic	29.8%

The detection rate percentages quoted here for *Stiller Integrity Master* (*IM*), are included only for continuity's sake really. The product is not an anti-virus scanner – it is primarily an integrity checker. As such it does perform a scan of a system prior to building its checksum database. Since the virus scanner is only a minor part of the *IM* product, updates are not frequently available, and thus detection rates are not high.

To compare the results directly to those obtained for the other products would be equivalent to comparing apples to oranges. The results may be of interest to some of our readers who may use *IM* however, hence their inclusion.

Unsurprisingly, the relatively static arena of boot viruses is where *IM* performs best, detecting all the ItW boot viruses. Elsewhere in the test-sets where changes over the past year have been fast and furious, the percentages are lower, especially in the Macro and Polymorphic test-sets.

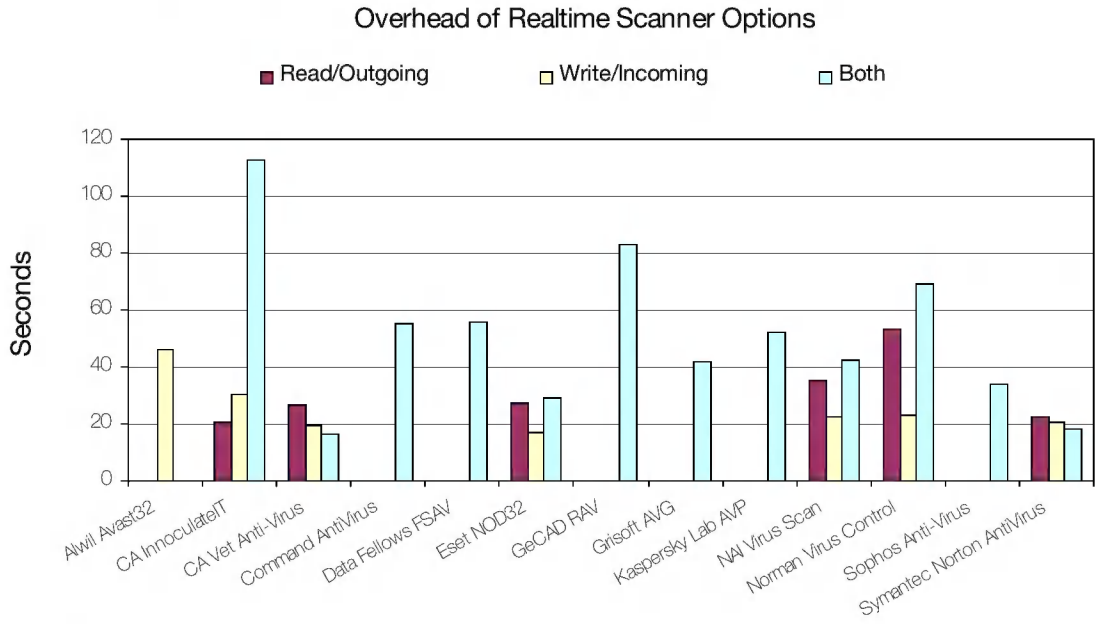
Symantec Norton AntiVirus v5.02.01

ItW Overall	99.8%	Macro	99.4%
ItW Overall (o/a)	99.8%	Standard	99.7%
ItW File	100.0%	Polymorphic	93.9%

Failure to implement complete detection of Win95/Fono infected boot sectors (as noted in previous reviews) in both on-demand and on-access scanning, once again prevents *Symantec's NAV* from claiming the VB 100% award.

Elsewhere, detection rates were admirable, and misses were few and far between. Samples infected with PP97M/Vic.A, W97M/Lys (B, C and D variants) were missed in the Macro set, and the only miss in the Standard set was an executable sample infected with {W95/W97M}/Heathen.A. The lowest detection rate was observed in the Polymorphic set, due to the product's failure to detect samples infected with the A and B variants of ACG.

Identical results were obtained for on-access scanning – a fact that few products can boast about. The overhead of the on-access scanner was the lowest out of all the products tested, whereas the on-demand scanning speed of *NAV* was fairly average, and in keeping with the bulk of the products. The *Bloodhound* heuristics employed by default in *NAV* did not register any false positives during scanning of the clean executable and OLE2 file sets.



	Hard Disk Scanning Speed					
	Executables			OLE2 files		
	Time (min:sec)	Throughput (kB/s)	FPs [susp]	Time (min:sec)	Throughput (kB/s)	FPs [susp]
Alwil Avast32	21:43	419.7	1	2:18	496.4	0
CA InnoculateIT	6:44	1353.8	0	0:28	2446.5	0
CA Vet Anti-Virus	10:20	882.1	[1]	0:17	4029.5	0
Command AntiVirus	6:27	1413.3	0	0:29	2362.1	0
Data Fellows FSAV	14:10	643.4	[3]	0:48	1427.1	0
Dialogue Science DrWeb32	17:42	515.0	1+[18]	0:52	1317.3	[1]
Eset NOD32	3:12	2848.6	0	0:23	2978.3	0
GeCAD RAV	23:04	395.2	[1]	1:02	1104.9	0
Grisoft AVG	14:29	629.4	10	0:23	2978.3	0
Kaspersky Lab AVP	3:50	2378.0	[2]	0:39	1756.4	0
NAI VirusScan	10:40	854.6	0	0:48	1427.1	0
Norman Virus Control	6:30	1402.4	0	0:34	2014.8	0
Proland Protector Plus	4:51	1879.5	5	n/t	n/t	n/t
Sophos Anti-Virus	11:20	804.3	0	0:33	2075.8	0
Stiller Integrity Master	5:33	1642.4	1+[47]	0:53	1292.5	1
Symantec Norton AntiVirus	8:41	1049.8	0	0:42	1631.0	0

For the 100th time...

Recent events have led us to believe that it is time we reminded ourselves exactly what the VB 100% award is all about. Just who is it designed to benefit? Does it provide the definitive standard to which products should aspire? The sole criterion of a 'good' product?

By simple definition, the VB 100% award is a certification scheme which identifies products capable of detecting all the viruses currently in-the-wild (as defined by the WildList Organisation) *at the time of testing*. The time dependancy of the award is fundamental to its usefulness. Unlike for some of the other certification schemes out there, recent WildLists are used for the VB 100% award. In this comparative, for example, products had to be submitted by 2 July, and the ItW testing was performed against a June WildList (which was announced in mid-June). The award logos which are issued to the appropriate vendors can quite justifiably be reproduced by the anti-virus developers as a marketing aid. By doing so, users familiar with the scheme can quickly spot products of good pedigree.

This last comment is important – of good pedigree. Not 'the best'. *Virus Bulletin* receives no end of enquiries as to the 'best' anti-virus product, from a variety of sources – both home users and corporates. The simplest yardstick by which to compare anti-virus products is detection rate. The VB 100% award gives an at-a-glance picture of products that did, and those that did not, 'make the grade' during the tests. Thus, following the results across a series of tests enables the leading products to be easily identified.

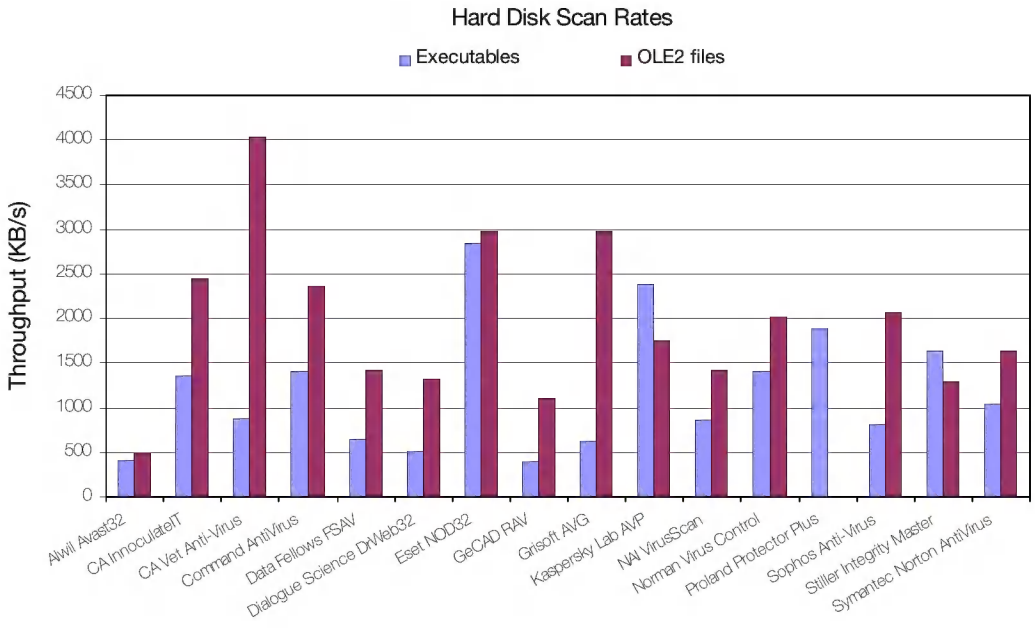
Whether looking from within the anti-virus circle or not, it is obvious that an awful lot of factors besides detection rates are important in selecting the most suitable product. A more accurate description of equipping oneself with virus protection might be to speak of it in terms of an anti-virus *service* – a package that in addition to the product itself, includes ongoing updates, technical support, and the like. The VB 100% award includes no measure of such factors, and as such is not itself a measure of the 'best' product.

The developers of each product obviously want to receive the VB 100% award for each test entered. The desire to do so has no doubt resulted in the improvement of many of the on-demand scanners. However, as with any certification process there is a danger in its over-emphasis. As mentioned above, it is a measure of only one aspect of a product's capabilities.

The anti-virus industry itself is partly responsible for the over-emphasis on certification schemes. The anti-virus marketing arena is an aggressive area, in which vendors do not pull any punches. Decorating products with the accolades of certifications A through Z is no doubt a successful

marketing tool. And why should this not be the case? Where earned, it is perfectly fair for products to bear the fruits of their labour.

Recently however, *Virus Bulletin* has noticed a couple of the anti-virus vendors displaying an altered VB 100% logo, one with the date removed. A marketroid's dream – an ageless certification scheme, once passed, forever qualified.



Besides being a breach of the conditions under which the award is handed out, more importantly, such an act fully intends to mislead the very people the VB 100% award is designed to help – users seeking genuine, impartial anti-virus advice.

In summary, the VB 100% awards are not by themselves an adequate summation of the entire results observed during a comparative review. Instead they provide the readers with a quick guide to the products which have been best kept up to date with changes in the virus scene, and they provide the vendors with a widely recognized mark of achievement.

Changes to the VB 100% Award

Since its introduction in January 1998, the VB 100% certification scheme has concentrated solely upon on-demand scanning. However, much anti-virus protection nowadays is centered upon on-access scanning. As such, the VB 100% certification scheme is evolving to incorporate on-access scanning as from the next comparative in the November 1999 issue.

Thus far, the VB 100% award has certainly been a success in that whilst striving to pass the regular certification tests, the anti-virus products have no doubt improved. With the inclusion of on-access scanning into the certification scheme as from the next comparative review, we hope this improvement will carry forth into the world of the on-access scanners, undoubtedly the weakest feature of the products in general.

Summary

Returning to the results presented in this review, it is clear that for most of the products tested, high detection rates across the board were observed. *Kaspersky Lab's AVP* steals the limelight with its detection of all the samples in both on-demand and on-access scanning. Close on its heels are *NOD32* from *Eset*, which detected all the ItW samples in on-demand and on-access tests, and *NAI's VirusScan* which but for its failure to detect extensionless samples, would also have achieved a clean sweep. Needless to say, both *AVP* and *NOD32* earn the VB 100% award this month. Four other products also managed to detect all the ItW viruses during on-demand scanning – *Norman Virus Control*, *GeCAD's RAV* and *Computer Associate's* brace of anti-virus products, *Vet Anti-Virus* and *InnoculateIT*. Interestingly, out of these six VB 100% clad products only *AVP* and *NOD32* manage to achieve the same standard during on-access scanning.

It is pleasing to note that *PowerPoint* file formats now seem to be supported by all the major products, at least in on-demand scanning. The same is not true of *Access* files, and so four of the sixteen products missed samples infected with A97M/Accessiv variants. Also surprising was the observation that password protected files, are still ignored by certain products. Thus *Word* document samples infected with W97M/Pwd.A were missed.

Technical Details

Test Environment: Server: *Compaq Prolinea 590*, 90MHz Pentium with 80 MB of RAM, 2 GB hard disk, running *NetWare 4.10*. Workstations: Three 166 MHz Pentium-MMX workstations with 64 MB RAM, 4 GB hard disks, CD-ROM and 3.5-inch floppy, all running *Windows NT* with *Service Pack 5* applied. The workstations could be rebuilt from image back-ups, and the test-sets were stored in a read-only directory on the server. All timed tests were performed on a single machine that was not connected to the network for the duration of the timed tests, but was otherwise configured identically to that described above.

Virus Test-sets: Complete listings of the test-sets used are at http://www.virusbtn.com/Comparatives/NT/199909/test_sets.html.

A complete description of the results calculation protocol is at http://www.virusbtn.com/Comparatives/Win95/199801/protocol.html.

SUBSCRIPTION RATES

Subscription price for 1 year (12 issues) including first-class/airmail delivery:

UK £195, Europe £225, International £245 (US$395)

Editorial enquiries, subscription enquiries, orders and payments:

Virus Bulletin Ltd, The Pentagon, Abingdon Science Park, Abingdon, Oxfordshire, OX14 3YP, England

Tel 01235 555139, International Tel +44 1235 555139
Fax 01235 531889, International Fax +44 1235 531889
Email: editorial@virusbtn.com
World Wide Web: http://www.virusbtn.com/

US subscriptions only:

Virus Bulletin, 18 Commerce Way, Woburn, MA 01801, USA

Tel (781) 9377768, Fax (781) 9320251

END NOTES AND NEWS

Scotland-based *Calluna Technology* announces the launch of *PC BODYGUARD Lab Edition*, a single card which plugs into an ISA expansion slot. Aimed at small businesses, the *Hardwall* technology within the product protects against accidental damage or malicious intervention (i.e. Internet viruses) in the corporate environment. *Calluna* claims that the product massively reduces PC maintenance costs. *PC BODYGUARD Lab Edition* retails at £90 excluding VAT. For more details contact *Calluna*; Tel +44 700 2255862 or visit the Web site http://www.calluna.com/.

***Data Fellows F-Secure Workstation Suite v4.0* is shipping now.** Based on a three tier, data security management *F-Secure* architecture, the new product's centralized management functionality provides automated upgrades and updates. Furthermore, it combines anti-virus protection with strong encryption for network traffic and files stored on the hard disk. For more information contact; Tel +1 408 9386700, fax +1 408 9386701 or email Tracey.Thomas@DataFellows.com.

CompSec'99, the 16th World Conference on Computer Security, Audit and Control will take place from 3–5 November 1999 at the QE2 Centre, Westminster, London, UK. A Directors' Briefing will be held on 4 November. Conference topics include malicious software, firewalls, network security and Year 2000 contingency planning. For more details contact Tracy Stokes at *Elsevier*; Tel +44 1865 843297, fax +44 1865 843958, or email t.stokes@elsevier.co.uk.

Following a recent email security breach at the British Houses of Parliament, ***Content Technologies Ltd* is encouraging the wide-spread corporate use of *MAILsweeper*, part of its *MIMEsweeper* family**. MAILsweeper can be used to identify and quarantine email containing viruses, hoaxes and malicious information. For more details, contact Catherine Jamieson; Tel +44 118 9301300 or email info@mimesweeper.com.

On 11 August – total eclipse day – the London Borough of Richmond's council offices were plunged into metaphorical darkness when they were sent, via email, a compiled batch file which deleted all unopened files using C:/windows/deltree.com. **The file was called ECLIPSE.COM** and its sender had used the name of the newly appointed (but not yet active) CEO of the offices. Police are currently investigating disgruntled employees and ex-employees.

The *Computer Security Institute's* 26th annual conference and exhibition is to be held from 15–17 November 1999 at the Marriott Wardman Park Hotel in Washington DC. For more information on the 85 featured presentations or pre- and post-conference seminars, contact *CSI*: Tel +1 415 9052626 or visit http://www.gocsi.com/.

In Brussels, Belgium, from 4–7 March 2000, **the ninth annual *EICAR* conference**, also known as the first European Anti-Malware Conference, takes place. For more information, to place a booking or to order a timetable visit the Web site at http://www.eicar.dk/.

An Advanced Internet Security workshop on 25 October 1999 is to be followed by a two day Implementing Windows NT Security course at the *Sophos* training suite in Abingdon, UK. For further information, or to reserve your place, contact Daniel Trotman; Tel +44 1235 559933, fax +44 1235 559935, or visit the company Web site http://www.sophos.com/. *Sophos Anti-Virus* has been selected as a finalist for Computing magazine's 1999 Awards for Excellence. *Sophos* is one of five finalists in the Network Software category, and the only anti-virus developer to be shortlisted. Final judging and the announcement of the winners will take place on 6 October 1999 at the Grosvenor House Hotel, London.

Microsoft* has acknowledged a security hole in its software *Jet which can allow code contained in an *Excel 97* worksheet, hidden in a Web page or sent via email, to plant viruses, delete data or read files. The problem only applies to *Jet v3.51*, which shipped with *Office 97*. All customers are advised to upgrade immediately to *Jet v4.0* via *Microsoft's* OfficeUpdate Web site. *Jet* is used by several *Microsoft* products including its *Exchange* messaging server. It is also the default database used with the company's popular Visual Basic development tool. The recently released *Office 2000* already uses *Jet v4.0* and is unaffected by the hole.

The ninth annual Virus Bulletin Conference
30 September & 1 October
Vancouver, Canada
http://www.virusbtn.com/.